Packet Signing 기법을 활용한 SCTP 방식의 RADIUS 프로토콜

김영세, 김기천*
건국대학교 IT융합정보보호학과, 건국대학교 컴퓨터공학과*
rladudtp87@konkuk.ac.kr, kckim@konkuk.ac.kr*

The RADIUS protocol using SCTP Packet Signing Method

Youngse Kim, Keecheon Kim*
Dept. of IT Convergence Information Security, Konkuk University
Dept. of Computer Science & Engineering, Konkuk University*

요 약

Remote Authentication Dial-In User Service (RADIUS)은 원격지의 사용자들을 인증하기 위한 대표적인 프로토콜이다. 기존의 RADIUS인증은 인증 서버가 클라이언트 IP, URI정보와 같은 데이터 통신의 기본사항만을 확인한 후 세션을 설정한다. 하지만, 악의적인 공격자가 IP 혹은 URI정보를 위/변조해서 인증을 요청하거나 반복적인 인증요청을 통한 DOS(Denial Of Service)공격을 수행할 경우 부적절한 인증, 사용자의 인증시간 지연 및 실패와 같은 심각한 문제가 발생할 수 있다.

본 논문에서는 클라이언트와 서버가 공유하는 데이터의 Hash값을 검증한 후, Packet Signing 방식을 활용해서 유효한 사용자에게만 서버접근을 허용하는 방안을 제안한다.

1. 서론

RADIUS은 Authentication, Authorization, and Accounting (AAA) 프레임워크 기반의 인증 프로토콜로서 사용자들이 특정 네트워크/시스템에 접속을 요청할 때, 인증 서버에서 사용자의 ID, 패스워드 등의 정보를 검증한 후 인증 및 식별 작업을 수행한다. 본 논문에서는 기존의 Challenge Handshake Authentication Protocol(CHAP) 인증 방식을 응용하여 클라이언트가 Hash값을 생성한 후, 서버가 해당 값을 검증하는 방안을 제안한다.

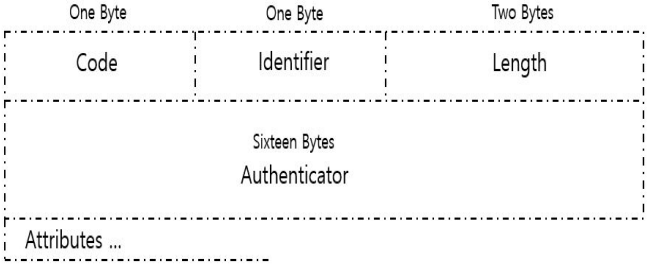
본 논문에서 제안한 방식을 통해서 Multi-Streaming 방식의 사용자 인증과정에서 메시지가 위/변조되는 문제를 개선하였으며, 성능 평가결과 속도면의 측면에서 Hash값 검증에 소요되는 시간은 사용자가 인지하지 못할 정도로 매우 짧지만, 제안한 방식을 통해서 기존 인증방식 보다 보안강도를 높일 수 있었다.

2. RADIUS 인증 방식 분석

2.1 RADIUS 프로토콜

RADIUS은 AAA 프레임워크 기반의 인증 프로토콜로서 사용자들이 특정 네트워크/시스템에 접속을 요청할 때, 인증 서버에서 사용자의 ID, 패스워드 등의 정보를 검증한 후 인증 및 식별 작업을 수행한다. 인증 서버는 수신한 식별정보를 근거로 적합한 사용자일 경우에는 원격접속을 허가하고, 그렇지 않을 경우에는 인증실패 메시지를 클라이언트에게 반환한다[7].

그림 1은 RADIUS 프로토콜의 Packet Format을 보여주고 있다[1]. 1 Byte 크기의 Code 필드는 RADIUS 인증 서버에서 가장 먼저 체크하는 필드로서 해당 값이 누락되거나 부적합 할 경우, 별도의 프로세스 없이 해당 패킷을 폐기 처리 하는데 이 과정을 'silently discard'라고 한다. Identifier 필드는 1 Byte 크기가 할당되고 packet의 requests와 replies를 매칭 하는 정보를 제공한다. 16 Bytes 크기의 Authenticator필드는 RADIUS 서버로부터의 응답을 인증하고 Password Hiding 알고리즘에서 사용되는 필드이며 Request Authenticator, Response Authenticator의 2종류가 있다.



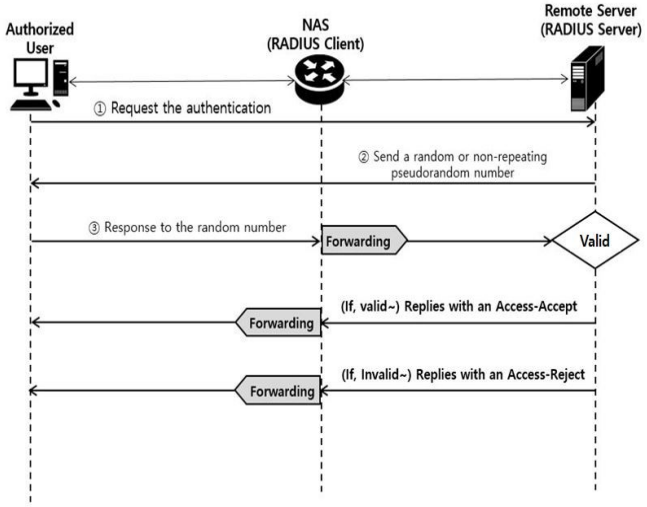
(그림 1) RADIUS 프로토콜의 Packet Format

* 교신저자

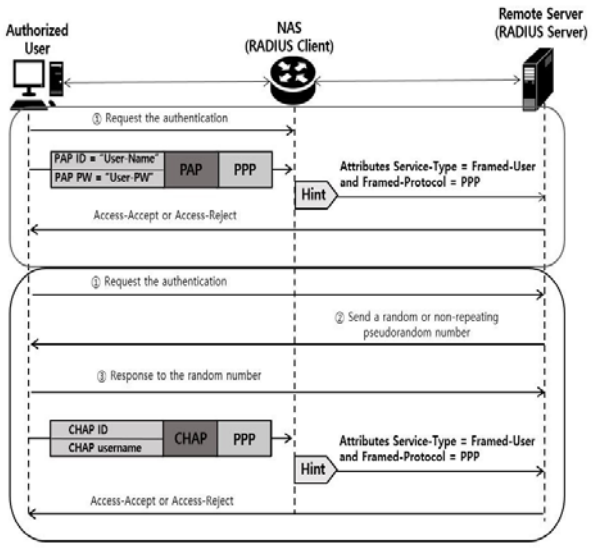
2.2 RADIUS 인증

RADIUS 프로토콜은 다양한 인증 방식을 지원하며, 일반적으로 Password Authentication Protocol(PAP)와 Challenge Handshake Authentication Protocol(CHAP) 인증 방식을 사용한다[3]. 두 인증방식은 모두 Point to Point Protocol(PPP) 인증용 프로토콜이다.

그림 2는 CHAP 인증의 기본구조인 Challenge & Response 동작 과정을 보여주고 있으며 그림 3은 PAP, CHAP의 인증과정을 비교해서 보여주고 있다.

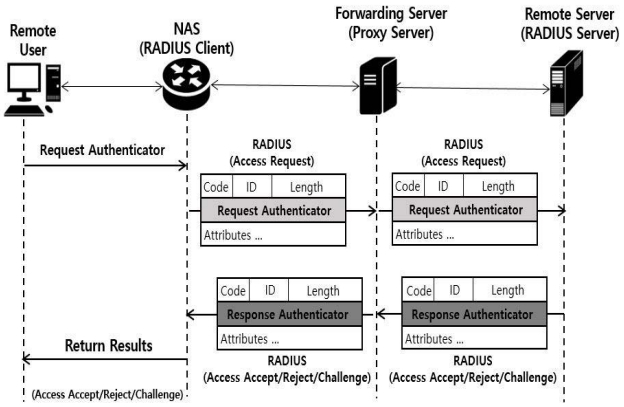


(그림 2) Challenge & Response 방식



(그림 3) PAP & CHAP 방식

RADIUS 프로토콜의 사용자 인증과 권한부여과정은 최초 클라이언트가 인증요청을 할 때, RADIUS 서버에 의해서 동시에 수행되고 자원체크는 사용자의 네트워크 접근이 허용된 이후에 자원사용 시작/종료시점 중간에 주기적인 상태 업데이트를 통해서 이뤄진다[1]. 그림 4는 RADIUS 프로토콜의 사용자 인증, 권한부여 과정을 보여주고 있다.



(그림 4) 사용자 인증, 권한부여 과정

2.3 SCTP방식을 활용한 인증

Stream Control Transmission Protocol(SCTP)는 데이터 통신을 위한 전송계층 프로토콜로서 혼잡제어를 통해 경로장애 복구기능을 제공하는 'multi-homing', 하나의 Association을 설정하여 다양한 종류의 데이터를 여러 개의 스트림으로 분리해서 독립적으로 전송할 수 있는 'multi-streaming'의 특징이 있다[2].

SCTP는 송/수신 객체간의 associations을 설정한 후, 데이터 전송을 위한 메인 경로를 지정한다. 그 이외의 다른 경로들은 예비 경로로 지정하고, 메인경로에 장애 발생시 대체경로로 활용한다. 또한 여러 stream에서 동시에 데이터를 전송할 수 있는데, 특정 stream에서 전송이 지연될 경우 다른 stream에 영향을 주지 않고 독립적으로 데이터를 전송한다[4].

3. 이슈 사항 분석

3.1 Multi-Streaming 방식의 사용자 인증

SCTP는 단일 SCTP Association을 설정한 후, 다수의 stream을 각 경로별로 전송할 수 있다. 송/수신 객체간의 associations을 설정한 후, 데이터 전송을 위한 메인 경로를 지정하고 특정 stream에서 전송이 지연될 경우 다른 stream에 영향을 주지 않고 독립적으로 데이터를 전송한다.

해당 Multi-Streaming기능을 통해서 대용량 데이터를 효율적으로 제어할 수 있다. 또한, 각 Stream 경로별로 Stream Sequence Number(SSN)를 유지해서 전송도중 특정 경로에서 오류가 발생할 경우, 해당 경로를 필터링하고 다른 경로를 선택하는 장애복구 기능이 있다[4]. 따라서 여러 사용자의 RADIUS인증 요청을 신속하게 처리할 수 있다.

3.2 각 경로별 전송 데이터의 무결성 유지

RADIUS가 SCTP방식을 사용해서 Device간의 인증절차를 수행할 경우, 여러 사용자의 인증요청을 신속하게 처리할 수 있다. 하지만 각 경로별 전송 데이터의 무결성을

유지해야만 송/수신 객체간의 신뢰를 확보할 수 있다.

만약, 악의적인 공격자가 IP 혹은 URI정보를 위/변조해서 인증을 요청하거나 반복적인 인증요청을 통한 Dos(Denial Of Service)공격을 수행할 경우 부적절한 인증, 사용자의 인증시간 지연 및 실패와 같은 심각한 문제가 발생할 수 있다[5].

Multi-Streaming 방식의 사용자 인증과정에서 확인할 수 있듯이, 악의적인 공격자에 의해서 메시지가 위/변조될 가능성이 있으며, SCTP기반 RADIUS의 기존 프로세스는 사용자 인증 처리에 대한 추가적인 과정이 필요하다고 분석되었다.

4. 개선된 SCTP 방식의 RADIUS 프로토콜

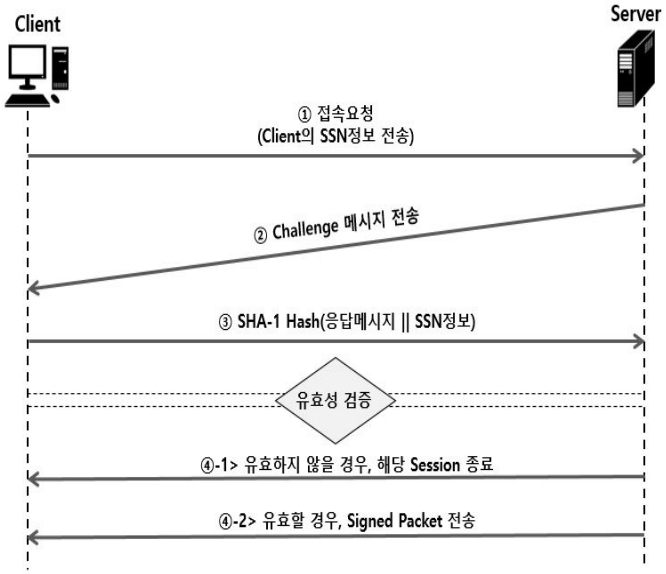
4.1 Packet Signing 방식의 사용자 인증

본 논문에서는 클라이언트와 서버가 공유하는 데이터의 Hash값을 검증한 후, Packet Signing 방식을 활용해서 유효한 사용자에게만 서버접근을 허용하는 방안을 제안한다[6].

사용자 인증처리 과정은 다음과 같다.

(1) 클라이언트는 특정 IP의 서버로 접속을 시도한다.
 - 클라이언트는 자신의 SSN정보를 서버에 전송한다.
(2) 서버는 클라이언트에게 Challenge 메시지를 전송한다.
 - eg.> 현재 시간정보는?
(3) 클라이언트는 Challenge 메시지에 대한 응답메시지에 SSN정보를 첨부한 후, SHA-1 Hash처리해서 서버로 전송한다.
(4) 서버는 클라이언트가 최초 접속시도를 할 때, 전송한 SSN정보와 Challenge의 적절한 응답 값을 자체적으로 계산한다.
(5-1) 서버는 클라이언트가 전송한 Hash값을 검증한 후, 유효하지 않을 경우 해당 Session을 종료한다.
(5-2) 클라이언트가 전송한 Hash값이 유효할 경우 서버는 해당 Session이 안전함을 보증하기 위해서, 클라이언트에게 Signed Packet을 전송한다.

그림 5는 Packet Signing 방식을 활용한 사용자 인증처리과정을 보여주고 있다.



(그림 5) Packet Signing 방식을 활용한 인증 처리 과정

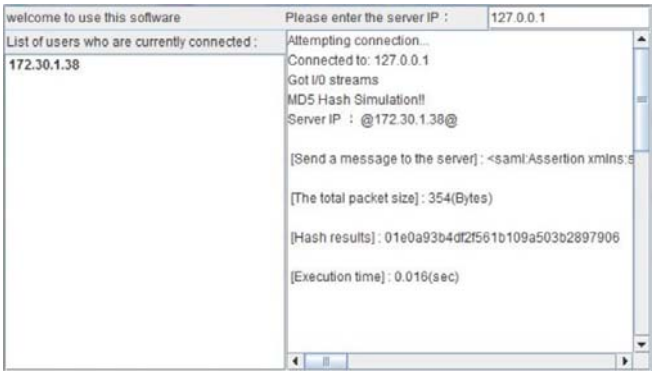
4.2 성능 평가

본 논문에서 제안하는 방식의 시뮬레이션을 위한 시스템 사양은 다음과 같다.

- CPU : Intel(R) Core(TM) i5-4590, 3.3GHz
- RAM 8G
- HDD 500G
- OS : Windows 7 professional k 64bit version

시뮬레이션 내용은 다음과 같다.

(1) SCTP 방식의 스트림 서버를 구축한다.
(2) 클라이언트는 접속하고자 하는 서버의 IP주소를 입력한다.
(3) 서버의 multi-streaming기능을 적용하여 요청하는 클라이언트는 5명으로 지정한다.
(4) 서버는 클라이언트의 요청정보를 검증한 후, 접속을 허가한다.
(5) 클라이언트가 Hash값을 생성한 후, 서버가 해당 값을 검증하는 시간을 측정한다.

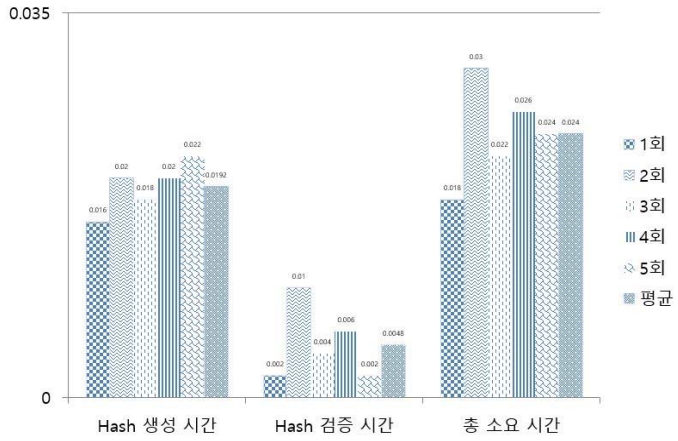


(그림 6) 클라이언트의 Hash값 생성과정

(그림 7) 서버의 Hash값 검증처리 과정

<표 1> 시뮬레이션 결과

단위 (sec)	1회	2회	3회	4회	5회	평균
Hash값 생성 소요 시간	0.016	0.02	0.018	0.02	0.022	0.0192
Hash값 검증 소요 시간	0.002	0.01	0.004	0.006	0.002	0.0048
총 소요 시간	0.018	0.03	0.022	0.026	0.024	0.024



(그림 8) 시뮬레이션 결과

기존의 RADIUS인증은 인증 서버가 클라이언트 IP, URI정보와 같은 데이터 통신의 기본사항만을 확인한다. 하지만, 다수의 사용자가 동시에 인증을 요청하고 재인증이 빈번하게 발생하는 환경에서는 추가적인 사용자 인증 처리과정이 요구된다.

앞서 살펴본 CHAP인증 방식을 응용하여 클라이언트는 응답메시지에 SSN정보를 첨부한 후, Hash처리해서 전송한다. 그 후에 인증 서버가 전송된 Hash값을 검증한 후, 인증여부를 결정한다.

그림 6, 그림 7은 각각 클라이언트의 Hash값 생성과정 및 서버의 Hash값 검증처리 과정을 보여주고 있다. 테스트 결과, 추가된 인증과정에 소요되는 평균시간은 0.024초로 측정 되었으며 속도면의 측면에서 기존 인증과정과 큰 차이가 없음을 알 수 있다.

따라서 클라이언트가 Hash값을 생성한 후, 서버가 해당 값을 검증하는 절차를 Multi-Streaming기능을 사용하는 SCTP방식의 RADIUS인증과정에 추가하는 것이 보안성을 높이면서 효율적인 프로세스가 될 수 있음을 알 수 있다.

5. 결론 및 향후 연구 방향

본 논문에서는 SCTP방식의 RADIUS 프로토콜 인증방식을 분석한 후, Multi-Streaming환경에서 각 경로별 전송 데이터의 무결성을 유지하는 방안을 연구했다. Hash값 검증에 소요되는 시간은 사용자가 인지하지 못할 정도로 매우 짧지만, 제안한 방식을 통해서 기존 인증방식 보다 보안강도를 높일 수 있었다.

다수의 사용자가 여러 디바이스를 사용해서 동시에 인증을 요청하고 재인증이 빈번하게 발생하는 환경에서는 송/수신되는 데이터의 무결성을 유지하기 위해서 인증주체가 객체를 보증하기 위한 수단이 필요하다. 따라서 인증주체와 객체가 상호간에 공유한 정보를 기반으로 생성된 Hash값을 인증절차에 활용한 후, 최종적으로 인증 서버가 서명한 패킷을 클라이언트에게 전송하는 방식을 고려해 볼 수 있다.

향후 연구과제로는 Cloud 환경에서 효율적으로 Private Data와 Public Data를 구분해서 전송하기 위한 방안으로 Diameter 프레임워크 기반의 SCTP전송방식을 활용해서 연구하고자 한다.

Acknowledgement

이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 정보통신 기술진흥센터의 지원(No.B0511-16-0001, 글로벌 딜리버리 클라우드 플랫폼의 대규모 OTT 서비스 적용을 위한 방송•통신 사업자 공동의 시범 사업)으로 수행된 연구임

참고문헌

[1] C. Rigney, S. Willens, A. Rubens, W. Simpson. “Remote Authentication Dial In User Service (RADIUS)”. RFC 2865. June 2000

[2] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko. “Diameter Base Protocol”. RFC 3588. September 2003

[3] B. Aboba, J. Wood. “Authentication, Authorization and Accounting (AAA) Transport Profile”. RFC 3539. June 2003

[4] R. Stewart, Ed. "Stream Control Transmission Protocol". RFC 4960. September 2007

[5] P.Venkadesh, Julia Punitha Malar Dhas, S.V.Divya. "Techniques to Enhance Security in SCTP for Multi-Homed Networks". Proceedings of 2015 Global Conference on Communication Technologies(GCCT 2015)

[6] L. Peterson, B. Davie, R. van Brandenburg, Ed. "Framework for Content Distribution Network Interconnection (CDNI)". RFC 7336. August 2014

[7] Youngse Kim, Keecheon Kim, "The RADIUS protocol Improved packet encryption and transmission method of Larger packets" The 2015 Fall Conference of the korea information processing society.